PATENT

SETTLEMENT OF AUCTIONS BY DETERMINING QUANTITY BEFORE PRICE

Inventors: Kang Le Yang
Kevin Ka-Shun Fung
Daniel Ming-Chiu Chan
Michael Kwok-Po Ng

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0001] This invention relates generally to the settlement of auctions and, more particularly, to the settlement of auctions by determining the fill of orders (i.e., the quantity to be filled) before setting price.

2. Description of the Related Art

[0002] A variety of financial instruments, which shall be referred to generally as contracts, are currently traded in many different markets. These contracts can take a variety of forms and can be related to a variety of activities or events. For example, the contracts could range from options and futures to betting. Participants in the markets typically make offers to buy contract(s) (also known as bids) and/or offers to sell contract(s) (also known as offers). Each bid and offer typically has a price limit associated with it. The participants in the market could include individual participants, financial intermediaries, and/or market makers, such as brokerage houses or banks.

[0003] Furthermore, the buyers and sellers can be either short or long. For example, a long seller is a seller already having a position in the market and holding the contract for which the seller makes an offer. A short seller is a seller who does not yet have ownership of the contract being offered for short sale. Similarly, a buyer may be making a bid to cover a contract previously offered for sale.

[0004] Typically, the interaction between the market participants can take place via three conventional structures: conventional order matching, conventional market making, and conventional auctions. In conventional order matching, offers to buy and sell are centralized, typically in an exchange, and orders are filled by matching them with the complementary order. Individual participants can then buy or sell until an equilibrium for a particular contract is reached. Typically, the exchange takes no risk in the market. In conventional market making, a market maker takes a position opposite to other market participants. Thus, a market maker may sell or buy contracts to other market participants.

[0005] In conventional auctions, a contract is typically offered for sale to any market participant. Conventional auctions can take a variety of forms. In certain conventional auctions, the contract is initially offered at a high price. The price is progressively lowered until a bid is made and the contract is sold. In conventional Dutch auctions, offers to buy are accumulated and the lowest price necessary to sell the entire lot of contracts becomes the price at which the contracts are sold. Mutualized risk pools can also be considered a form of auction. In a mutualized risk pool, market participants can choose to put money into a pool up until a cutoff time. When the pool closes, the money placed into the pool determines the prices of the contracts.

[0006] The relationships between buyers, sellers, and those who facilitate the market (e.g., market makers, exchanges, auction organizers, etc.) can be complex. For example, in the case of betting, individuals are usually limited to the role of making bets (i.e., purchasing contracts) and the role of selling contracts is the province of the bookmaker. Furthermore, unnecessary uncertainty may be created in these relationships, which indirectly increases trading costs.

[0007] Although conventional structures allow transactions to take place and for the market to come to equilibrium, conventional structures also have drawbacks. For example, conventional structures may not result in a high degree of liquidity. Generally speaking, high liquidity is desirable. A higher liquidity allows the market participants to move in and out of the market more easily. In addition, exchanges desire a high liquidity because exchanges typically obtain a profit based upon the number of transactions carried out. Higher liquidity usually translates into a higher number of transactions and therefore a greater profit for the exchange. Market makers desire a higher liquidity because a high liquidity translates to a higher number of transactions, lower risk for the market maker and a lower cost of borrowing capital for the market maker. Thus, it is desirable for a higher liquidity in the market place than may be available using the conventional structures.

[0008] Conventional order matching, market making and auctions also have other drawbacks. Conventional order matching often does not function well when there is an insufficient number of sellers that actually have contract(s) to sell, as opposed to a short seller. For example, there may be a disproportionate number of buyers without matching sellers. As a result, there will be lowered liquidity. In some situations, conventional market makers may actually have an incentive to reduce the competitive nature of the marketplace because the market maker may act to their own advantage, rather than to the advantage of the market as a whole. Conventional auctions can take time to organize and identify the winner(s) and, in the absence of sufficient offers, can fail to facilitate trading.

[0009] Moreover, conventional auctions generally can only be used in trading of single products but not related products. An example would be the Dutch auction where a single product of a fixed quantity is being sold, and the price of the lowest qualifying bid would be used as the price. As it stands, the Dutch auction cannot be applied to trading of financial products based on a continuous variable (e.g., an underlying stock price).

[0010] Accordingly, there is a need for mechanisms that increase the liquidity of markets and/or address the drawbacks of conventional auctions.

SUMMARY OF THE INVENTION

[0011] One aspect of the present invention overcomes the limitations of the prior art by settling an auction of contracts as follows. The orders collected during the auction form an auction pool. The fill for orders in the auction pool is established without setting an auction settlement price (ASP) for the contracts in the auction pool. The fill determines a qualified pool of orders and an unqualified pool. The ASPs for contracts in the qualified pool are set based on the fill. Both the process for establishing the fill for orders and the process for setting the ASPs preferably are based on LP and/or QP optimizations. This can increase the speed of these processes.

[0012] In one aspect of the invention, the fill is established according to price aggressiveness of the orders. For example, LP optimization can be used to maximize an aggregate price aggressiveness subject to constraints defined in the orders (e.g., quantity limits) and subject to the requirement that filled orders form complete sets (as further described below). Dummy zero fill orders can also be used to maximize price aggressiveness, as explained in further detail below. For orders of the same price aggressiveness, the fill can be determined according to other priority factors, such as a time chop.

[0013] In another aspect of the invention, the ASPs are set by first determining a lower price bound and an upper price bound for contracts, according to price constraints defined in the orders and further according to the fill for the orders. The ASPs are selected to fall within the range defined by the lower and upper price bounds. The price bounds can be selected in a number of ways. For example, the price gap between the lower and upper price bounds can be maximized, the variance of the price bounds from some other price figure can be minimized and/or the price constraints can be taken from only qualified orders, only unqualified orders, or a combination of qualified and unqualified orders. Similarly, different methods can be used to select the ASPs within the price range. Different types of interpolation, including linear interpolation, can be used.

[0014] In yet another aspect of the invention, orders for contracts are expressed in terms of a special set of contracts, which shall be referred to as basic units. The fills and ASPs are established relative to the basic units, which in turn will establish the fills and ASPs for the original contracts.

[0015] Other aspects of the invention include different variations to address different types of contracts, and devices and systems corresponding to all of the above.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] The invention has other advantages and features which will be more readily apparent from the following detailed description of the invention and the appended claims, when taken in conjunction with the accompanying drawings, in which:

[0017] FIG. 1 (prior art) is a table showing different types of contracts.

[0018] FIGS. 2A-2E are payoff diagrams illustrating different complete sets of contracts.

[0019] FIGS. 3-4 are payoff diagrams illustrating different sets of basic units.

[0020] FIGS. 5A-5C are payoff diagrams illustrating the decomposition of a call spread into basic units.

[0021] FIG. 6 is a timeline of an auction according to the invention.

[0022] FIG. 7 is a flow diagram showing various methods for settling an auction according to the invention.

[0023] FIG. 8 is a flow diagram showing one method for establishing the fill.

[0024] FIG. 9 is a flow diagram showing one method for setting the auction settlement price.

[0025] FIG. 10 is a diagram illustrating the fill for a numerical example.

[0026] FIG. 11 is a block diagram of a computer system suitable for use with the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0027] This invention relates generally to buying and selling contracts, particularly in the context of auctions. It is useful to begin with some underlying concepts, as illustrated in FIGS. 1-5.

[0028] FIG. 1 is a table showing the following types of contracts: discrete, digital call, digital put, digital range, call spread and put spread. These examples are selected for illustrative purposes; the invention is not limited to these contracts. In the table, the payoff diagram describes the payoff of a contract. The x-axis in the payoff diagram is the underlying event (or state) upon which the contract depends and the y-axis is the payoff. The payoff equation is a mathematical description of the payoff. The payoff function for a contract Cx shall be denoted as P(Cx). The columns "Payoff Diagram" and "Payoff Equation" are graphical and mathematical representations of P(Cx) for the specific contracts shown.

[0029] Some general remarks can be made about these contracts. First, note that the underlying event can be modeled as either discrete or continuous. It is discrete for the discrete contract and continuous for the other examples. The event can be modeled as discrete if there are a relatively small, finite number of possible outcomes. In the table, the possible outcome states are denoted as x1, x2, etc. It is more appropriately modeled as continuous if it is continuous or there are a large enough number of possible outcomes that it can be effectively modeled as continuous. In the table, the payoffs for the continuous cases depend on the value of state x. Note that events can be modeled as either discrete or continuous, depending on the content of the contract. In addition, the underlying events can be multi-dimensional, not just one-dimensional as shown in the table.

[0030] One example of a discrete event is a horse race, where the possible outcomes are x1: horse 1 wins, x2: horse 2 wins, etc. Another example is a sporting event where the possible outcomes are x1: Challenger wins the America's Cup or x2: Defender wins the America's Cup (assuming no ties). Another example is the stock market, with possible outcomes x1: certain index closes below Z or x2: at or above Z. Alternately, the outcomes might be x1: index closes below Z, x2: in the range between Z and Z+100, x3: in the range Z+100 to Z+200, x4: in the range Z+200 to Z+300, or x5: above Z+300 (ignoring boundaries for now).

[0031] One example of a continuous event is the price of an index. Another example would be the time margin of victory of horse 1 (although that example is not frequently used). A third example is the difference between the NFC champion's score and the AFC champion's score in the Super Bowl.

[0032] For continuous events, it can be important to handle boundary conditions correctly so that aberrations do not appear. In FIG. 1, discontinuous boundaries (such as for the digital put, digital call and digital range) are handled by taking the average of the two values. In this way, if the digital put and digital range contracts were combined, the payoff at x=a would be the notional N. In contrast, if the payoff function = N at all boundaries, then combining these two contracts would yield a payoff of 2N at the boundary x=a, which would be an undesirable anomaly. The boundaries can be handled in many ways. For example, the entire notional N can be allocated to one or the other of the contracts, so long as the sum was still N.

[0033] Also note that the payoff function can be either digital or not digital. A digital payoff function is one that can take on only two possible values (excluding boundary conditions). For example, the discrete, digital put, digital call and digital range contracts all have digital payoff functions: either 0 or the notional N. The two spreads have proportional payoffs, at least in the range of $a<x<b$. The call spread has a diverging slope (i.e., payoff increases as x increases) and the put spread has a converging slope (i.e., payoff decreases as x increases).

[0034] FIGS. 2A-2E illustrate the concept of complete sets. When someone holds a contract Cx, his payoff for that particular contact is P(Cx). When someone holds a set of contracts, the aggregate payoff for that set of contracts is the sum of the individual payoffs for each contract. A complete set of contracts is a set of contracts for which the aggregate payoff is a constant. This constant shall be referred to as the settlement value (SV) of the complete set. Mathematically, a set of contracts {Cn} is a complete set if

$$\sum P(Cn) = SV \text{ for all possible outcomes} \qquad \text{(Eqn. 1A)}$$

where the summation is over all contracts n in the set. Graphically, the payoff diagram for a complete set is a constant line at payoff SV. Since the payoff diagram of a complete set is a constant value SV, the "fair market" price of a complete set (ignoring time value of money and transaction costs for now) is also SV. In other words,

$$\sum \text{Price}(Cn) = SV \qquad \text{(Eqn. 1B)}$$

where Price(Cn) is the price of contract Cn. The actual prices for the contracts may vary.

[0035] FIG. 2A shows a complete set for the discrete case. The possible outcomes are x1, x2, etc., which are assumed to be mutually exclusive and collectively exhaustive in this example. In other words, one and only one of the outcomes x1, x2, etc. must occur. The complete set is the set of contracts Cx1, Cx2, etc. covering all possible outcomes where each of these contracts has the same notional. In this way, regardless of which outcome occurs, the holder of the complete set will receive a payoff equal to the notional. It is like betting on both black and red in roulette (assuming no 0 or 00). As another example, if the event is a horse race, then the set of wagers that pays \$10 if horse 1 wins, \$10 if horse 2 wins, etc. is a complete set.

[0036] Complete sets can be formed in more than one way. FIG. 2B shows a different complete set for the discrete case. In this example, the complete set contains two contracts: Cx1 and C(not x1). C(not x1) pays the notional if any of x2, x3, etc. occur. Note that the contract C(not x1) is equivalent to the set of contracts {Cx2, Cx3, etc.} and the payoffs and fair market prices are related:

$$P(C(\text{not } x1)) = P(Cx2) + P(Cx3) + \ldots \qquad \text{(Eqn. 2A)}$$

$$\text{Price}(C(\text{not } x1)) = \text{Price}(Cx2) + \text{Price}(Cx3) + \ldots \qquad \text{(Eqn. 2B)}$$

[0037] Combining Eqns. 1B and 2B yields

$$\text{Price}(Cx1) = SV - \text{Price}(C(\text{not } x1)) \qquad \text{(Eqn. 3A)}$$

In other words, an offer to sell Cx1 at price Z or higher is equivalent in aggressiveness to an offer to buy C(not x1) at price (SV-Z) or lower. These concepts can be used to convert between short and long positions, as will be discussed in more detail below.

[0038] FIG. 2C is an example of a complete set for a continuous case. In this example, the complete set includes a digital put with upper bound x = 260, three digital ranges that cover the range 260 < x < 320 in 20 point increments, and a digital call with lower bound x = 320. All contracts have the same notional. Regardless of the value of x, the holder of the complete set will be entitled to receive a payoff equal to the notional. Note that the boundary conditions should be handled properly so that the payoff diagram of the complete set does not vary from the notional at the boundaries.

[0039] FIG. 2D is an example of a complete set formed by overlapping contracts. The boundaries are slightly shifted so the different contracts can be more clearly seen. In this case, the complete set includes three contracts: a digital put with upper bound 300 and notional N, a digital call with lower bound 280 and notional N, and a contract that is not(digital range of 280/300). This last contract has two pieces. The sum of the three contracts is a complete set with SV = 2 N. In this example, the contracts are not mutually exclusive. For example, an outcome of x = 282 will result in a payoff for both the digital call and the digital put.

[0040] FIG. 2E is an example of a complete set using non-digital payoff functions. In this example, a call spread and a put spread form a complete set. Both spreads are proportional in the range 260 < x < 280 and have the same notionals. Thus, the sum of the payoff diagrams is a constant with settlement value equal to the notional.

[0041] FIGS. 2A-2E are merely examples. Complete sets can be formed in many different ways. However, it is important to recall that a complete set of contracts will pay the settlement value regardless of the outcome of the underlying event. Thus, the fair market price for the complete set is its settlement value. As an example, assume three possible outcomes x1, x2 and x3. Further assume that each of the corresponding contracts Cx1, Cx2, Cx3 has a notional of N = $100. Then the set of three contracts Cx1, Cx2, Cx3 forms a complete set with SV = $100. Accordingly, if there is an opportunity to buy Cx1 for $50, Cx2 for $30 and Cx3 for $19 (or a total of $99), this is an opportunity to buy the complete set, which is worth $100, for only $99. It is a $1 arbitrage opportunity. Conversely, if there is an opportunity to sell Cx1 for $51, Cx2 for $31 and Cx3 for $19 (or a total of $101), this is an opportunity to sell a $100 complete set for $101. These are simple examples to illustrate the point. More complex examples, including taking advantage of the long/short equivalence described above, will be described in more detail below.

[0042] Put in another way, if a market organizer (or anyone else) buys just Cx1, it will be taking some contract risk depending on whether x1 occurs or not. However, if the organizer buys or sells a complete set at price SV, the organizer will not be taking any contract risk because the two are equivalent. Thus, the organizer can buy and sell complete sets for other purposes, for example to increase the liquidity of the market or reduce credit risk. For further examples of how this can be accomplished, see co-pending U.S. Patent Application Serial No. 10/601,173 (attorney docket no. 2700P), entitled "Method and System for Utilizing a Special Purpose Vehicle for Improving the Liquidity of Transactions," filed on June 20, 2003; and co-pending U.S. Patent Application Serial No. 10/600,903 (attorney docket no. 2701P), entitled "Method and System for Managing Credit-related and Exchange Rate-related Risk," filed on June 20, 2003; all of which are incorporated by reference herein.

[0043] The discussion thus far has also neglected certain other effects. For example, the time value of money has not been factored into the discussion. A payoff of $100 that occurs a year from now (e.g., when the underlying event is resolved) is not really worth $100 today. The discussion also does not account for transaction costs. However, these and other factors can be handled using conventional techniques.

[0044] Standard present value concepts can be used to account for the time value of money, allowing direct comparison of dollar figures. For example, Eqn. 1B states that the fair market price of a complete set should be the settlement value. However, if the payoff will occur one year from now, then, more precisely, the fair market price today of a complete set should be the settlement value, as measured in dollars one year from now, discounted by the annual interest rate. Or if the equation is cast in future dollars, then the fair market price today of a complete set, after interest for a year, should be equal to the settlement value, measured in future dollars.

[0045] Similarly, amounts can be adjusted up or down, as necessary, in order to account for transaction costs, profit or other effects. For example, if a minimum profit Z is desired, then the sum of the prices of the contracts in the complete set should be (SV-Z) or lower to yield profit Z for the buyer, or (SV+Z) or higher to yield profit Z for the seller. In the interest of clarity, all examples will continue to neglect these factors, with the understanding that they can be handled using conventional techniques. For further examples and discussion, see co-pending U.S. Patent Application Serial No. 10/600,137 (attorney docket no. 2626P), entitled "Method and System for Improving the Liquidity of Transactions," filed on June 20, 2003; co-pending U.S. Patent

Application Serial No. 10/601,173 (attorney docket no. 2700P), entitled "Method and System for Utilizing a Special Purpose Vehicle for Improving the Liquidity of Transactions," filed on June 20, 2003; and co-pending U.S. Patent Application Serial No. 10/600,903 (attorney docket no. 2701P), entitled "Method and System for Managing Credit-related and Exchange Rate-related Risk," filed on June 20, 2003; all of which are incorporated by reference herein. Note that in these referenced patent applications, the term "initial settlement value" is used to refer to the settlement value when the events occur (i.e., when the contracts mature) and "settlement value" is used to refer to other time frames, so that time discounting may be used to equate the current settlement value with the initial settlement value.

[0046] As shown in FIGS. 2A-2E, complete sets can be formed in many different ways. In order to facilitate the formation of complete sets, it can be useful to define a special set of contracts, which shall be referred to as basic units. In many markets, the contracts traded in that market can take many different forms. They can also be quite complex. It can be difficult to try to assemble these "native" contracts into complete sets. As an alternative, the native contracts can be decomposed into an equivalent combination of basic units and complete sets are then formed using the basic units.

[0047] There can be many choices for the set of basic units. However, in many cases it is advantageous for the basic units to be digital with the same notional, and mutually exclusive and collectively exhaustive. In this case, it will be simpler to decompose the native contracts into basic units, and it will also be simpler to form the basic units into complete sets. Specifically, for any given contract, there will only be one possible decomposition into basic units. In addition, there will only be one way to form a complete set, and that is by including one of each basic unit. The settlement value of the complete set will equal the notional. For each basic unit, the event where the basic unit pays the notional shall be referred to as the state of the basic unit. The set of basic units preferably should also have the correct shapes so that all possible native contracts can be decomposed into basic units. While these characteristics are preferred, they are not required.

[0048] In the discrete case of events x1, x2, etc., one good choice of basic units is Cx1, Cx2, etc. where all contracts have the same notional. The states sn for the basic units Cx1, Cx2, etc. are then s1=x1, s2=x2, etc. This set of basic units is both mutually exclusive and collectively exhaustive. Furthermore, in many discrete cases, the "native" contracts are simply single event contracts (e.g., Cx5) so that each native contract is a basic unit. In cases where more complex native contracts are used, the decomposition into basic units is straightforward.

[0049] FIG. 3 shows a set of basic units for a case where the underlying event x is continuous but the payoff diagrams are all digital. The set of basic units is the digital put with boundary 260, the set of 6 digital ranges shown, and the digital call with boundary 320. All basic units have the same notional. This set of basic units can be used to decompose any native contract that has a digital payoff with boundaries at 260, 270, . . . 320.

[0050] The set of basic units can change over time. For example, if x approaches the upper end of the range, the set can be expanded to include more digital ranges at the high end. In addition, digital ranges at the low end can be decommissioned, with a corresponding change in the digital put. Alternately, if finer ranges are desired, the current digital ranges can be subdivided, for example using digital ranges with a span of 2 units instead of 10. The spans also are not required to be identical. When making these changes, it is preferable to select a new set of basic units that is backwards compatible with the old set.

[0051] FIG. 4 shows a set of basic units that are based on digital calls and digital puts, all with the same notional. Note that this set is not mutually exclusive. However, it can also be used to decompose any native contract that has a digital payoff with boundaries at 260, 270, . . . 320.

[0052] FIG. 5A illustrates the decomposition of a call spread into the following basic units: a diverging slope of 100/110 (the triangular region marked A), a digital range of 110/120 (rectangle B), a diverging slope of 110/120 (C), two digital ranges of 120/130 (D), a diverging slope of 120/130 (E), and three digital calls at 130 (F). All basic units have the same notional, although they are not all digital. More generally, if corresponding converging and diverging slopes are added to the set of basic units shown in FIGS. 3 or 4, then call spreads and put spreads can also be decomposed into basic units.

[0053] FIG. 5B shows a set of basic units that can be used to decompose the call spread in FIG. 5A. This set includes a digital put 100, a digital call 130, three digital ranges spanning 100/130, three converging basic units spanning 100/130, and three diverging basic units spanning 100/130. This set of basic units can be used to decompose any of the continuous contracts shown in FIG. 1, so long as the boundaries match those of the basic units (i.e., a and b are 100, 110, 120 or 130). Note that in addition to the complete set Eqns. 1-3 above, the relationships of the basic units also imply additional constraints. For example, the digital range 100/110 is equivalent to the sum of a converging 100/110 and a diverging 100/110. Therefore,

Price(converging 100/110) + Price(diverging 100/110) = Price (range 100/110)(Eqn. 3B)

Thus, within this group of three basic units, there are really only two independent basic units with respect to fair market price. If the fair market prices of two of the basic units are known, then the fair market price of the third can be calculated by Eqn. 3B. An analogous situation exists for other relationships.

[0054] FIG. 5C shows an alternate set of basic units. Rather than using the triangular shaped converging and diverging basic units, a series of rectangular "sub" basic units is used to approximate the proportional payoff diagram. FIG. 5C shows the approximation of a diverging 100/110. The converging 100/110 would be approximated by the complimentary set of rectangular units, so that the two sets would sum to the notional. In many cases, the underlying variable x is not truly continuous. For example, it may have a tick value that is the smallest increment of x. In this case, the rectangular "approximation" becomes exact if the width of the rectangles equals the tick value. In one implementation, the "sub" basic units are chosen to have the same notional, so the rectangular unit closest to 100 may include one sub basic unit while the taller rectangular unit closest to 110 may include many sub basic units to reach the desired height.

[0055] The above are merely examples. For further examples of how to select a set of basic units, see co-pending U.S. Patent Application Serial No. 10/600,137 (attorney docket no. 2626P), entitled "Method and System for Improving the Liquidity of Transactions," filed on June 20, 2003 and incorporated by reference.

[0056] The concepts of basic units can be important because they allow markets with many different types of contracts to be decomposed into basic units, which can then be used to facilitate the formation of complete sets. All of the following examples will be illustrated using discrete contracts with only a few possible outcome states, typically three to five, where the states are assumed to be mutually exclusive and collectively exhaustive. This is done for clarity. Otherwise, the examples will involve so many numbers as to obscure the point being illustrated. However, by using the concept of basic units, these simple examples can be extended to complex markets with many different types of contracts.

[0057] Let us now turn to auctions. FIG. 6 is an example timeline of an auction. At time 410 (2:00 pm in this example), the auction opens and begins accepting orders for contracts. Typically, the auction organizer will define the types of contracts that are supported by the auction. The auction continues for some period of time and then closes 420 (at 3:00 pm in this example), after which orders are no longer accepted. In the specific example of FIG. 6, there is also a tail period 415 where orders can no longer be withdrawn. At some point, the auction is settled, meaning that the price of contracts is determined and each pending order (i.e., the auction pool) is either filled or not. The events underlying the contract typically occur after the auction both closes 420 and settles.

[0058] Settlement of the auction can occur in a number of ways. FIG. 7 is a flow diagram showing various ways of settling an auction according to the invention. In many conventional approaches, the final prices (i.e., the auction settlement prices or ASPs) for different contracts are set first, and then orders in the auction pool are either filled or not by comparing the price limits in the orders with the ASPs. In contrast, in FIG. 7, the fill for orders in the auction pool is established 710 without setting the ASPs. The fill defines a qualified pool of orders and an unqualified pool. The ASPs for orders in the qualified pool are then set 750 based on the previously established fill.

[0059] The interior of boxes 710 and 750 illustrate specific examples of these two steps. In this particular example of step 710, the fill is established 720 according to a price aggressiveness of the orders. Orders which are more price aggressive are given priority over less aggressive orders. In this context, price aggressiveness depends on price constraints or price limits in the orders but does not depend on the ASP since the ASP has not been set yet. In one approach, orders are filled so that an aggregate price aggressiveness for orders in the qualified pool is maximized. If some orders are equal in price aggressiveness, then other priority factors, such as time chop, can be used to determine 722 the fill among these orders. Some additional adjustment 724 may also be desirable, for example to ensure that all fills are an integer number if that is a requirement of the auction or to meet other constraints in the orders. Throughout steps 720-724, the process of establishing which orders are filled also attempts to form complete sets from the filled orders. Once the fill is established 710, one approach to step 750 first establishes 760 price bounds for the various contracts based on the price constraints in the orders in the qualified pool. The ASPs are then set 762 within these bounds, for example by interpolation between upper and lower price bounds. Additional adjustment 764 may also take place.

[0060] FIGS. 8-9 give further variations and examples of steps 710 and 750, respectively. The mathematical explanations are based on the following assumptions, in order to simplify the examples. These assumptions are not requirements. All contracts and orders are expressed in terms of basic units. The basic units are digital with the same notional, and they are also mutually exclusive and collectively exhaustive. For each basic unit, the event where the basic unit pays the notional is referred to as the state of the basic unit. All orders are assumed to be buy orders and may have a price limit and/or a quantity limit. The following nomenclature is used:

N notional, which is assumed to be 100

S total number of states (or basic units)

j = 1, · · · , S state index

n total number of orders, all assumed to be buy orders

i = 1, · · · , n order index

[πi,qi,ci] description of order i, where

πi price limit for the order (contracts to be purchased for not more than πi per unit of contract)

qi quantity limit for the order (not more than qi units of contract to be purchased)

ci vector of length S, describing the contract ordered, in terms of the basic units

cij the number of basic unit j contained in one unit of contract i

xi fill quantity for order i (xi units of order i will be filled)

zj fill quantity for the zero fill order for basic unit j (as explained below)

p vector of length S, with the ASP for each basic unit (the ASP for contract ci can be determined by taking the inner product of vectors p and ci)

pj ASP for basic unit j

M number of complete sets formed

For example, the order [60,100,(1 1 0)] is an order to buy up to 100 units of a contract, where one unit of the contract consists of a combination of one unit of state 1 and one unit of state 2. One unit of state 1 is the contract that pays a notional $100 if state 1 occurs, and one unit of state 2 is the contract that pays a notional $100 if state 2 occurs. This order also sets a price limit of $60 for one unit of the contract. Note that (1 1 1) is a complete set with a settlement value of $100. Zero fill orders are single state buy orders with price limit 0 and unlimited quantity. For example, the zero fill order for state 3 is [0,∞,(0 0 1)]. Their utility will be explained below.

[0061] FIG. 8 and the following text describe various methods for establishing 710 the fill using LP optimization techniques, price aggressiveness and the notion of complete sets. One advantage of the specific mathematical formulations shown below is they are all LP optimizations, which is numerically efficient to solve. The method begins by performing 820 an LP optimization that maximizes an aggregate price aggressiveness, subject to constraints defined in the orders and also subject to forming complete sets. Mathematically, one formulation of the optimization is

$$\begin{aligned} &Max \quad \sum_{i=1}^{n}\pi_i x_i - 100M \cdots\cdots\cdots AGG \\ &s.t. \quad \begin{cases} x_i \le q_i & i = 1 \ldots n \\ \sum_{i=1}^{n} c_{ij}x_i + z_j = M & j = 1 \ldots S \end{cases} \end{aligned} \qquad \text{(Optimization 1)}$$

The objective function is the aggregate price aggressiveness (AGG), which inherently contains information about the ASP p. The variables in this optimization are M (the number of complete sets) and xi and zj (the fill quantities of the orders and of the zero fill orders). The first constraint is that the fill quantity xi for order i cannot exceed the quantity limit qi in order i. The second constraint is that the total quantity filled for basic unit j must equal M for all basic units. In other words, the filled quantities form a total of M complete sets.

[0062] The AGG can be interpreted as the "profit over market" which would be realized if the filled quantities were all sold at their price limit (as opposed to the ASP). "Market" price for a complete set is the notional, which is 100 in this example. Put in another way, during the optimization, there will be a current qualified pool and current unqualified pool. If any orders in the unqualified pool can be combined to form a complete set with a price limit over 100, then

AGG can be increased by moving these orders to the qualified pool. Similarly, if any orders in the qualified pool can be combined to form a complete set with a price limit under 100, then AGG can be increased by moving these orders to the unqualified pool. Thus, this optimization tends to select orders that can be combined into complete sets at prices over the notional.

[0063] In step 822, the fill for zero fill orders is maximized, subject to the same constraints in Optimization 1 and also subject to maintaining the AGG determined in Optimization 1. Mathematically, one formulation is

$$Max \ \sum_{j=1}^{S} z_j \cdots\cdots\cdots\cdots\cdots\cdots\cdots \ Z$$

$$s.t. \begin{cases} x_i \le q_i & i = 1 \ldots n \\ \sum_{i=1}^{n} c_{ij} x_i + z_j = M & j = 1 \ldots S \\ \sum_{i=1}^{n} \pi_i x_i - 100M = AGG & \end{cases} \qquad \text{(Optimization 2)}$$

This optimization further enforces priority by price aggressiveness.

[0064] For example, consider an auction that contains the following orders: [30, 1, (1 0 0)], [75, 1, (0 1 0)], [75, 1, (0 1 1)]. The combination of one unit each of [30, 1, (1 0 0)] and [75, 1, (0 1 1)] will form a complete set that contributes 105-100=5 to the AGG. The combination of one unit each of [30, 1, (1 0 0)], [75, 1, (0 1 0)] and the zero fill order [0,∞,(0 0 1)] will also form a complete set that contributes 105-100=5 to the AGG. The second combination should be favored because [75, 1, (0 1 0)] is more price aggressive than [75, 1, (0 1 1)]. However, Optimization 1 does not distinguish between these two combinations. Optimization 2 does, and will select the more price aggressive combination since it increases the fill for the zero fill orders.

[0065] After fixing the aggregate price aggressiveness (AGG) and the fill for the zero-fill orders (Z), the orders are considered sequentially according to priority (other than price aggressiveness) and each order's fill will be maximized 824 in turn, subject to the fills of the previously considered orders. This can be used to break "ties" between orders that are equal in price aggressiveness.

[0066] For example if order 1 is [30, 100, (1 0 0)] and order 5 is [30, 200, (1 0 0)], the two orders are equal in price aggressiveness and Optimizations 1 and 2 will not distinguish between the two. If 150 units are to be allocated to these two orders, some other priority can be used to do the allocation. Time chop (i.e., time of receipt of the order) is one example. As other examples, certain size orders (e.g., either larger orders or smaller orders) may be preferred over others, or orders from certain customers may be given priority over others. Other priority factors can be used. If time chop is used and order 1 is received before order 5, then the fills will be x1 = 100 and x5 = 50.

[0067] One formulation is to consider each of the orders i in order of priority and to establish the fill xk for the current order being considered by

$$\begin{aligned}
&Max \quad x_k \\
&s.t. \begin{cases}
x_i = X_i & i = \text{indices of orders with higher priority} \\
x_i \le q_i & i = \text{indices of orders with lower priority} \\
\sum_{i=1}^{n} c_{ij} x_i + z_j = M & j = 1 \ldots S \\
\sum_{i=1}^{n} \pi_i x_i - 100M = AGG & \\
\sum_{j=1}^{S} z_j = Z &
\end{cases}
\end{aligned} \qquad \text{(Optimization 3)}$$

The optimization is run once for each order. Xi is the fill determined in previous Optimization 3's for orders with higher priority. The last three constraints are from the previous optimizations.

[0068] The previous optimizations may result in non-integer fills, which can be undesirable in some cases. Step 826 eliminates non-integer fills. One method to avoid non-integer fills is to use integer programming, but this typically will slow down the settlement process.

Alternately, non-integer results can simply be rounded off to integer values, but the rounding may cause the formation of incomplete sets, which typically increases the risk for the auction holder.

[0069] Another approach is based on LP optimization. The fill for any order contains an integer part and a decimal part. For example, if fill x1=327.5, then the integer part is 327 and the decimal part is 0.5. In addition, the fills resulting from the above optimizations form many complete sets. The problem of non-integer fills can be addressed by removing complete sets until all orders have integer fills. For example, consider the three orders: [90,100,(2 1 0)], [40,100,(0 1 0)] and [50,101,(0 0 1)]. The fills are 50.5, 50.5 and 101, respectively, forming a total of 101 complete sets. Note that 0.5 unit of order 1, 0.5 unit of order 2 and 1 unit of order 3 also form a complete set. If this complete set is removed from the original fills, then the resulting fills are 50, 50 and 100, which are all integers. The non-integer fill results from insufficient demand in some orders and the ratio of orders forming a complete set not being equal to all ones. For the above example, the orders form complete sets in the ratio of 1:1:2.

[0070] The following is one formulation.

$$\begin{aligned}&Min\quad M'\\&s.t.\begin{cases}Decimal\ x_i \le x'_i \le q_i & i=1...n\\ Decimal\ z_j \le z'_j \le z_j & j=1...S\\ \sum_{i=1}^{n} c_{ij}x'_i + z'_j = M' & j=1...S\end{cases}\end{aligned} \qquad \text{(Optimization 4)}$$

where xi and zj are the fills from the previous optimizations, and Decimal xi and Decimal zj are the decimal parts of those fills. The variables in the optimization are M' (the number of complete sets to remove), and x'i and z'j (the corresponding units of orders and zero fill orders, respectively). After reduction, the fills are new xi = old xi - x'i and new zj = old zj - z'j. In some cases, the optimization may be iterated or otherwise adjusted in order to eliminate the non-integer fills.

[0071] FIG. 8 is merely an example. Other variations will be apparent. For example, different objective functions can be used for the optimizations, techniques besides LP optimization can be used, or different formulations or approaches can be used to address the same problems. Even if the exact formulations shown above are used, not all of the steps need be used and they may be used in combination with additional other steps. For example, if all fills are integer or if non-integer fills are acceptable, then step 826 need not be used. Alternatively, if non-integer fills is an issue, it can be addressed by forcing integer behavior at some other point in the process. As another example, steps 822 and 824 also may not be needed in certain cases, for example if the corresponding problems are not present.

[0072] As another example, steps besides those shown can also be used and/or the steps can be performed iteratively. For example, adjustments of the fill can occur at any point in the process and adjustments for factors other than non-integer fill can be implemented. One example is that some orders may have a lower quantity limit. In other words, the order may be valid only if at least some minimum number of units can be purchased. Mathematically, either lower quantity limit ≤ xi ≤ qi or xi = 0. This constraint can be implemented by using the approach of Optimization 4, but replacing the lower bound of "Decimal xi" with a quantity that depends on the lower quantity limit for the order. For example, if the fill of order 4 is determined to be 100 but order 4 has a lower quantity limit of 500, then Optimization 4 can be run with the constraint x'4 = 100. The optimization will attempt to remove complete sets with 100 x4, thus fulfilling the requirement of a minimum order of 500 by not filling order 4. Alternately, if the initial fill determination indicates that order 4 will not be filled because it is allocated 100 units but requires a minimum of 500 units, then the entire fill determination can be iterated but removing order 4 from the auction pool for subsequent iterations (or, equivalently, by force x4 = 0 in Optimization 1 and subsequent optimizations).

[0073] Returning to FIG. 7, once the fills are established 710, the ASPs are then set 750. The fills from step 710 can be expressed by xi and zj. The fill step 710 also determines a qualified pool and an unqualified pool. In one approach, the qualified pool is defined as orders that are filled (i.e., xi > 0) and the unqualified pool is defined as orders that are not filled (i.e., xi = 0). The qualified and unqualified pools can also be defined in other ways. FIG. 9 and the corresponding text describe various ways for setting 750 the ASP. One advantage of the specific mathematical formulations is they are all LP or QP optimizations. As with FIG. 8, these are merely examples.

[0074] The qualified and unqualified pools impose price bounds on the states and, in one approach, some or all of these price bounds are used to set the ASPs. For qualified orders, the ASPs of the states should not violate the price limits of the orders. Mathematically, the price bounds imposed by qualified orders i are

$$\begin{aligned}&\sum_{j=1}^{S} c_{ij} u_j \leq \pi_i \\ &\sum_{j=1}^{S} (1\text{-}c_{ij}) l_j \geq 100\text{-}\pi_i\end{aligned} \qquad \text{(Eqn. 4)}$$

where lj and uj are the lower and upper price bounds for basic unit j. The second inequality is the buy-sell reflection of the original orders (assuming that the maximum value of all cij is 1). The ASP for basic units should satisfy both the buy side and sell side.

[0075] Unqualified orders also impose price bounds. The ASPs of the states should not make unqualified orders tradable. Mathematically, for unqualified orders i, the constraints corresponding to Eqn. 4 are

$$\begin{aligned}&\sum_{j=1}^{S} c_{ij} u_j \geq \pi_i \\ &\sum_{j=1}^{S} (1\text{-}c_{ij}) l_j \leq 100\text{-}\pi_i\end{aligned} \qquad \text{(Eqn. 5)}$$

These constraints are especially useful when there are no qualified orders and an indicative price is desired.

[0076] FIG. 9 shows one example for setting ASP based on maximizing the upper price bound and minimizing the lower price bound of each state, where the maximizations and minimizations are performed separately for upper and lower price bounds and also separately for each state. In other words, if a first optimization determines that the upper price bound u1 = 85, that value is not fixed for subsequent optimizations that find l1, u2, l2, u3, l3, etc. In step 960, the upper price bound of basic unit j is maximized. One formulation is

$$
\begin{aligned}
&Max\ u_j \\
&s.t\ \begin{cases}
x_i > 0 \Rightarrow \sum_{j=1}^{S} c_{ij} u_j \le \pi_i \\
x_i > 0 \Rightarrow \sum_{j=1}^{S} (1\text{-}c_{ij}) l_j \ge 100\text{-}\pi_i \\
l_j \le u_j \\
\sum_{j=1}^{S} l_j \le 100 \\
\sum_{j=1}^{S} u_j \ge 100
\end{cases}
\end{aligned}
\tag{Optimization 5}
$$

The first two constraints are the price limits imposed by the qualified pool (Eqn. 4). The qualified pool is used in the following examples, but this is purely for illustrative purposes. Any combination of constraints from the qualified and unqualified pools can be used. The third constraint requires that the lower price bound must be less than or equal to the upper price bound. The last two constraints are added as a safety check, as they should be automatically satisfied. The sum of the ASPs should equal 100 (the notional). Requiring that the sum of the lj ≤ 100 and the sum of the uj ≥ 100 is a check that the ASPs will fall between the lj and the uj. This is preferred since the ASPs can then be determined by interpolation, rather than by extrapolation. Extrapolation is more unpredictable and more likely to yield unacceptable results.

[0077] In step 961, the lower bound is found in a similar fashion. The formulation corresponding to Optimization 5 is:

$$Min \ l_j$$

$$s.t \quad \begin{cases} x_i > 0 \Rightarrow \sum_{j=1}^{S} c_{ij} u_j \leq \pi_i \\ x_i > 0 \Rightarrow \sum_{j=1}^{S} (1\text{-}c_{ij}) l_j \geq 100\text{-}\pi_i \\ l_j \leq u_j \\ \sum_{j=1}^{S} l_j \leq 100 \\ \sum_{j=1}^{S} u_j \geq 100 \end{cases} \qquad \text{(Optimization 6)}$$

[0078] Once the upper and lower price bounds are determined, a price within that range is determined. One approach uses linear interpolation 962 between the upper and lower price bounds. One formulation is

$$p_j = \frac{l_j(\sum_{k=1}^{S} u_k - 100) + u_j(100 - \sum_{k=1}^{S} l_k)}{\sum_{k=1}^{S} (u_k - l_k)} \qquad \text{(Eqn. 6)}$$

These prices reflect the relative demand of different states and this formulation automatically ensures that the sum of prices pj equals 100 (the notional). However, the resulting prices pj may violate the price constraints imposed by the orders. This is generally undesirable. As a result, prices can be adjusted 964 so that they no longer violate these price constraints. In one approach, the prices determined by Eqn. 6 are used as an initial starting point, and new prices are determined by minimizing the variance from these initial prices while requiring that price constraints in the orders be met. One formulation is

$$Min \sum_{j=1}^{S} (p_j - p'_j)^2$$

$$s.t \begin{cases} x_i > 0 \Rightarrow \sum_{j=1}^{S} c_{ij} u_j \leq \pi_i \\ x_i > 0 \Rightarrow \sum_{j=1}^{S} (1\text{-}c_{ij}) l_j \geq 100\text{-}\pi_i \\ \sum_{j=1}^{S} p_j = 100 \\ \sum_{j=1}^{S} z_j p_j \leq R \end{cases} \qquad \text{(Optimization 7)}$$

where p'j are the initial prices from Eqn. 6 and pj are the final ASPs. In this example, an additional fourth constraint is added. Recall that zj is the fill for zero fill orders. These orders do not correspond to any real orders in the auction pool. They represent a deviation of the qualified pool from complete sets. Hence, if they are priced at non-zero levels (which typically will be the case), then the auction holder or some other party will have some risk for these orders. In this optimization, the total risk for zero fill orders is limited to R.

[0079] An alternate approach determines the upper and lower price bounds by maximizing the sum of the price gaps for all states instead of individually maximizing each price gap for each state. A reference price h is introduced. The optimization will simultaneously try to minimize the variance of the price bounds from h. The reference price h can be a historical price, a price set by the auction organizer, the price from other price setting mechanisms or any other price. In this example, the maximization of the sum of the price gaps and the minimization of the variance relative to h is accomplished in one optimization, for example

$$Max \quad K*\sum_{j=1}^{S}(u_j-l_j)-\sum_{j=1}^{S}(u_j-h_j)^2-\sum_{j=1}^{S}(h_j-l_j)^2$$

$$s.t. \begin{cases} x_i>0 \Rightarrow \sum_{j=1}^{S}c_{ij}u_j \le \pi_i \\ x_i>0 \Rightarrow \sum_{j=1}^{S}(1\text{-}c_{ij})l_j \ge 100\text{-}\pi_i \\ l_j \ge 0 \\ l_j \le u_j \\ u_j \le 100 \\ \sum_{j=1}^{S} l_j \le 100 \\ \sum_{j=1}^{S} u_j \ge 100 \\ \sum_{j=1}^{S} z_j u_j \le R \end{cases} \qquad \text{(Optimization 8)}$$

The constant K can be used to adjust the relative importance of the price gap term and the variance terms. K will be large if the primary goal is to maximize the price gaps. Once the upper and lower price bounds are determined, the actual prices pj can be set as described previously.

[0080] Now consider a numerical example. Suppose there are three states and six orders:

Order 1:	[26,150,(1 0 0)]
Order 2:	[25,200,(1 0 0)]
Order 3:	[77,100,(0 1 0)]
Order 4:	[77,1200,(0 1 0)]
Order 5:	[80,500,(0 0 1)]
Order 6:	[20,300,(1 0 1)]

In this example, the auction is settled using the specific formulations described above.

[0081] First, the fill is established. Optimization 1 results in a maximum AGG = $37400 with a fill of 150, 200, 0, 500, 500, 0 for Orders 1-6 and a fill of 150 for the zero fill order for state 1. Optimization 2 is performed but does not change the result in this example. Optimization 3 allocates fill between Orders 3 and 4. It is possible to vary the fill of Order 3 and Order 4, as long as their sum of fills equal to 500, since they have the same shape and price limit. To make the fills uniquely determined, the fill is optimized according to their priority (time chop for example). Assume that Order 3 has higher priority than Order 4, then their fills are 100 and 400 respectively. There are no non-integer fills at this point, so Optimization 4 is skipped. The final fill is 150, 200, 100, 400, 500, 0 for Orders 1-6 and 150 for zero fill order 1. This result is shown graphically in FIG. 10.

[0082] Given this fill, ASPs can be determined in a variety of ways. For example, applying Optimizations 5 and 6 yields the price bounds shown in rows 1 and 2 of Table 1. Row 3 shows the linearly interpolated price as determined by Eqn. 6. Optimization 7 is performed to ensure that the price limits of qualified orders are not violated. Row 4 shows the results where the 150 zero fill orders for state 1 are limited to a total of $100 (or price of $0.67). This corresponds to R=100 in Optimization 7. Row 5 shows the ASPs without the risk constraint.

Table 1

	State 1	State 2	State 3
Upper Price Bound	$25.00	$77.00	$80.00
Lower Price Bound	0	0	0
Interpolated Price	13.75	42.30	43.95
ASP (with Risk Constraint)	0.67	48.84	50.49
ASP (without Risk Constraint)	13.75	42.30	43.95

[0083] As an alternate method for determining the ASPs, Optimization 8 can be applied instead. Alternately, the results shown in Table 1 can be used as the h values for Optimization 8. For this example, let h be (13.75, 42.30, 43.95), which are the ASPs in Table 1 without risk constraint. Also include the same risk constraint of R = $100 as before. The upper and lower price bounds are shown in Table 2. These are linearly interpolated to determine the ASPs, as shown in row 3.

Table 2

	State 1	State 2	State 3
Upper Price Bound	$0.67	$77.00	$80.00
Lower Price Bound	0.01	36.66	38.34
Interpolated Price	13.75	42.30	43.95
ASP (with Risk Constraint)	0.21	48.86	50.93

[0084] The examples given above used certain types of contracts and orders for clarity but the invention is not limited to these. For example, all orders were buy orders in the above examples. Sell orders can be handled in an analogous fashion. If interest and transactional costs are ignored, selling some contracts at a price p is equivalent to buying the complementary contracts at a price of N-p. For example, a sell order of [30,100,(1 0 0)] is equivalent to a buy order of [70,100,(0 1 1)]. If state 1 is the winning state, both orders will lose $70 per contract and if state 1 is not the winning state, both orders will gain $30 per contract. In one approach, if the auction pool contains a mixture of buy and sell orders, orders of one type can be transformed to equivalent orders of the other type for purposes of settling the auction.

[0085] As another example, orders may have constraints other than price or quantity. For example, some orders may limit price and total investment amount. There is a relationship between price, quantity and investment amount:

$$\text{Investment Amount} = \text{Quantity} \times \text{Price} \qquad \text{(Eqn. 7)}$$

Thus, an order that limits total investment amount and price can be modeled as an equivalent order with quantity and price limits. As an example, assume that an order has an investment limit of $1000 and a price limit of $25. This is equivalent to an order with a quantity limit of $1000/ASP and a price limit of $25. In all events, the bidder is willing to buy at least $1000 / $25 = 40 units. So, as a conservative estimate, this order can be modeled as an equivalent order with a quantity limit of 40 and a price limit of $25. If the settlement process results in an ASP of $10, which is significantly below the price limit of $25, then the process could be iterated with a new quantity limit of $1000 / $10 = 100 or something between 40 and 100. The price limit can also be reduced accordingly. Other variations will be apparent.

[0086] FIG. 11 is a block diagram of a system suitable for use with the present invention. Generally speaking, market participants 1110A-C and the auction organizer 1150 participate in the auction 1130 via a network 1120. The market participants 1110 enter their orders and receive updates about the auction via the network 1120. The auction organizer 1150 can also receive updates about the auction 1130 and can control the auction via the network 1120. In one specific embodiment, the network 1120 is the Internet, and the auction 1130 is hosted on a server 1132, with information stored on a database 1134. The market participants 1110 and the organizer 1150 access the Internet, typically by browsers such as Microsoft's Internet Explorer. The market participants can be individuals, but they can also include other entities, such as automatic trading programs. The server 1132 responds to requests from market participants 1110 and the organizer 1150.

[0087] It should be noted that FIG. 11 is simplified for clarity. For example, the roles of market participants 1110 and auction organizer 1150 can be implemented in a distributed fashion and/or divided among many different entities. The auction 1130 itself may also be distributed for redundancy and/or performance reasons. The server 1132 can contain different components, for example a fill module to determine 710 the fill, a price setting module to set 750 the ASPs, and an interface module to interface with outside entities (such as updates of stock prices, if relevant). Multiple servers, databases, load balancers, etc. can be used to implement the auction 1130.

[0088] As further clarification, the invention may be used with systems other than the Internet. For example, the various entities may communicate with each other over separate communications networks or dedicated communications channels, rather than through the common network 1120 of FIG. 11. Alternately, various parts of the system may be implemented by mobile components and may not be permanently attached to a communications network. For example, the different entities may interact via a wireless connection.

[0089] In alternate embodiments, the invention is implemented in computer hardware, firmware, software, and/or combinations thereof. Apparatus of the invention can be implemented in a computer program product tangibly embodied in a machine-readable storage device for execution by a programmable processor; and method steps of the invention can be performed by a programmable processor executing a program of instructions to perform functions of the invention by operating on input data and generating output. The invention can be implemented advantageously in one or more computer programs that are executable on a programmable system including at least one programmable processor coupled to receive data and instructions from, and to transmit data and instructions to, a data storage system, at least one input device, and at least one output device. Each computer program can be implemented in a high-level procedural or object-oriented programming language, or in assembly or machine language if desired; and in any case, the language can be a compiled or interpreted language. Suitable processors include, by way of example, both general and special purpose microprocessors. Generally, a processor will receive instructions and data from a read-only memory and/or a random access memory. Generally, a computer will include one or more mass storage devices for storing data files; such devices include magnetic disks, such as internal hard disks and removable disks; magneto-optical disks; and optical disks. Storage devices suitable for tangibly embodying computer program instructions and data include all forms of non-volatile memory, including by way of example semiconductor memory devices, such as EPROM, EEPROM, and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROM disks. Any of the foregoing can be supplemented by, or incorporated in, ASICs (application-specific integrated circuits) and other forms of hardware.

[0090] Although the detailed description contains many specifics, these should not be construed as limiting the scope of the invention but merely as illustrating different examples and aspects of the invention. It should be appreciated that the scope of the invention includes other embodiments not discussed in detail above. Various other modifications, changes and variations which will be apparent to those skilled in the art may be made in the arrangement, operation and details of the method and apparatus of the present invention disclosed herein without departing from the spirit and scope of the invention as defined in the appended claims. Therefore, the scope of the invention should be determined by the appended claims and their legal equivalents. Furthermore, no element, component or method step is intended to be dedicated to the public regardless of whether the element, component or method step is explicitly recited in the claims.